# Customized Multiprocessor Scheduling Algorithms for Real time Systems

Komal S. Bhalotiya
Department of Computer Science & Engineering G. H. Raisoni College of Engineering, Nagpur

## Abstract
For variety of reasons comprising instruction level parallelism, power consumption and memory speed mismatch, many of the chip manufacturers are moving towards the development of multicore processors. Multiprocessors are considered as a powerful computing resource because of their reliability and high performance.

Such Multiprocessor Real time system requires an efficient algorithm to determine when and on which processor a given task should execute. This paper presents a comparative study of different customized Multiprocessor scheduling algorithms which are for specific performance parameters and which maximizes the real time tasks that can be processed without violating timing constraints.

## Keywords
Real Time Operating System, Multi-Processor, Scheduling Algorithm**.**

## 1. INTRODUCTION
In today's Embedded system, Multi-Processor system are becoming the norm with increasing demand for higher performance under limited budgets. Real time systems are defined as those systems in which the correctness of system depends not only on the logical result of computation, but also on the time at the results are produced. If result is not obtained within limited time then result may be incorrect or no meaning of that result In such a type of system result must be obtained within the limited time constraints.Multi-Processors have appeared as a powerful computing means for running real time applications. An efficient scheduling algorithm is required to decide when and on which processor the given task should execute. Hence, the OS scheduler that assigns tasks to different processors is the crucial component for the performance of the real time systems on Multi-Processor. Scheduling is an important aspect in real time system to ensure soft/hard timing constraints. Real time systems make use of scheduling algorithms to maximize number of real time task that can be processed without violating timing constraints.

The performance of each and every scheduling algorithm depends on performance parameters. The following are some performance parameters of the Scheduling algorithms.

### 1.1 CPU Utilization
In the Multiprocessor Real time Operating System, CPU utilization is the parameter which is very important. All the processors in the system must be effectively utilized.

### 1.2 Task Migration
Task Migration is the performance parameter which measures the Number of context switches of the task on different processors.

### 1.3 Number of Preemption
It measures the number of preemptions of a task by a Higher Priority task.

### 1.4 Less Execution Time
This parameter measures the time taken by a scheduler for execution of a task.

### 1.5 Success Ratio
Success Ratio is the ratio of Number of tasks successfully scheduled to the Total Number of tasks arrived at the scheduler.

### 1.6 High Throughput
Scheduler must be able to achieve high Throughput.

### 1.7 Resource Utilization
In addition to the CPU Utilization, there should be the optimal use of system resources in Multi-Processor environment.

### 1.8 Effectiveness
Effectiveness of a Scheduler can be measured on any factor for example, Energy.

## 2. ALGORITHMS

### 2.1 Adaptive scheduling Algorithm
Adaptive scheduling algorithm is the combination of ACO(Ant colony Optimization) Scheduling Algorithm and EDF (Earliest Deadline First) Scheduling Algorithm for real time tasks on Multi-Processor environment. This adaptive algorithm is very useful when future workload condition of the system is unpredictable. This algorithm can automatically switch between EDF and ACO.During underload condition, the algorithm uses EDF Algorithm and priority of job will be decided dynamically depending on its deadline. During Overload condition, it uses ACO based Algorithm in which priority of the jobs will be decided depending on the pheromone value laid on each schedulable task and heuristic function.The performance of this algorithm is measured in terms of 2 performance parameters which are success ratio and Effective CPU utilization. In real time systems for measuring the performance of a scheduling algorithm the most important performance parameter is success ratio which determines whether the task is meeting its deadline. Success ratio is defined as the ratio of Number of tasks successfully scheduled to the total number of tasks arrived. In this algorithm effective CPU Utilization is considered as another performance parameter .Effective CPU Utilization is the ratio of summation of computation time of a task to the total time of scheduling.[2]

## 2.2 FPZL scheduling Algorithm

This algorithm is based on the state of zero laxity. Zero laxity is the state where execution time is equal to its deadline.

In this algorithm the task reaches a state of zero laxity it is given the highest priority.The priorities of all the tasks are considered as fixed until it reaches to a state of zero laxity. This algorithm is also known as a minimally dynamic algorithm because priority of a task can change at most once during its run time. Hence this algorithm bounds the number of preemptions which is very important performance parameter. The performance of this algorithm is measured in terms of Deadline analysis, CPU Utilization and response time performance parameters.[3]

## 2.3 IUF scheduling Algorithm

This algorithm targets the soft real time systems. It schedules the periodic tasks. This algorithm is based on instantaneous Utilization Factor [IUF]. Instantaneous utilization Factor [IUF] is the processor utilization of a task at any time instant. Here for every task after one time quantum the instantaneous utilization of task is computed. The task having highest instantaneous utilization is given the highest priority and task is given to the processor. The quantum for which task is applied to CPU is $Q_i$. The total sum of quantum of all task(for that quantum iteration only) is $\Sigma Q_i=Q$. Here,the Periodic task cycle(PTC) is computed for a given task set which is the LCM of period of invocation of all tasks. After computation of initially the CPU utilization is computed. CPU Utilization is the ratio of initial computation time and its initial period. Depending on this initial utilization the task having highest initial utilization is mapped to the CPU. At this point of time In a given PTC one task has executed for one time quantum.Now for calculating new instantaneous utilization factor agin the value of computation time and period is computed.For computing new computation time, the total sum of quantum of all task is subtracted from the previous computation time. In a similar way the new period is calculated. After this ,the new instantaneous utilization is computed. Likewise the process will be repeated.[3]

Using this algorithm we can easily compute that whether any task is going to miss its deadline or not which is a very critical performance parameter in real time systems.In addition to this the algorithm gives good schedulability, predictability and sustainability.

## 2.4 Modified IUF scheduling Algorithm

In many situations it is desirable to complete the important portion of every task rather than giving up completely the processing of some task. This is the aim of this algorithm to meet the deadline constraints. This is modified algorithm of a IUF scheduling Algorithm where it may be possible that some tasks may be given up completely by the scheduler In addition to that there is one more drawback in IUF scheduling algorithm that the context switching between tasks is very high. Hence this type of algorithm can only work in soft real time systems.

So in order to overcome the drawback of IUF scheduling algorithm and to reduce the context switching there was need to modify the IUF scheduling Algorithm.This algorithm was named by authors as "Modified Scheduling Algorithm". In this algorithm to schedule the important portion of every task,each taskis logically divided into 2 parts mandatory and optional. Mandatory portion is the portion which is the important portion of a task and which should always be executed.optional portion is the portion which is less important and can be given up by the scheduler.All mandatory portions of the tasks are scheduled according to the instantaneous utilization like IUF.In this algorithm first all the mandatory portions of every task is scheduled first and then the optional portions are considered for scheduling. For scheduling mandatory portion of every task first the CPU Utilization is computed.CPU Utilization is ratio of the initial computation time of task to their initial period.Rest of the procedure is same as that of IUF scheduling algorithm where after generating the CPU mapping for tasks,the mandatory portions of tasks are executed according to the highest instantaneous utilization. After scheduling all the mandatory portions the optional portions are to be scheduled. for this the shortest job First policy is employed.[4]

In this proposed algorithm authors have concluded that context switching, CPU Utilization and response time has been increased as compared to the IUF scheduling algorithm.

## 2.5 Incremental GA scheduling Algorithm

A novel Genetic Algorithm Scheduling Algorithm is having two unique features which distinguishes this GA from Traditional GA.First it uses flexible representation style which allows the GA to evolve both the structure and the value of the solutions.Because of above reason this GA possess the ability to identify and retail good building blocks.Second, this GA uses dynamically incremental fitness function which starts out rewarding for simpler goals, gradually increasing the difficulty of the desired fitness values or goals until a full solution is found. As a result, ourGAplaces no restrictions on the individuals that can be formed and does not require special operators or repair mechanisms to ensure validity.The modified versions of crossover and mutation operators are used.The advantage of this GA algorithm are that it is simple to use,requires minimal problem specific information and it is able to effectively adapt in dynamically changing environments[5].

# 3. COMPARISON BETWEEN DIFFERENT MULTI-PROCESSOR SCHEDULING ALGORITHM

The following table shows the comparative study between these customized scheduling algorithms for multi-Processor real time system on the basis of performance parameters concentrated.

**Table 1: Comparison between different Multi-Processor Scheduling algorithm on the basis of performance parameters considered**

| Algorithm | Performance parameter Concentrated |
|---|---|
| Adaptive Scheduling Algorithm | Effective CPU Utilization and Success Ratio |
| FPZL | Deadline Analysis, CPU Utilization and Response Time |
| IUF | Deadline analysis, CPU Utilization |
| Modified IUF | Number of preemptions, CPU Utilization, Response time |

| **Algorithm** | **Performance parameter Concentrated** |
|---|---|
| Incremental GA | CPU Utilization, Resource Utilization |

# 4. CONCLUSION

In this paper, some customized algorithms are reviewed and a comparative study has been done on the basis of performance parameters considered for measuring effectiveness of algorithms. As all the algorithms are for scheduling tasks on Multi-processor real time systems, one parameter which is common in all the algorithms for measuring and comparing their performance with other algorithm is CPU Utilization. In future, one adaptive algorithm can be developed which will be a combination of two or more algorithms for increasing the performance of a System. For Example, the FPZL and Modified IUF algorithm can be combined to make a new algorithm which will increase the performance of a Multi-Processor Real time system. Using this combination of algorithms maximum performance parameters can be covered and effectiveness of the system can be achieved.

# 5. REFERENCES

[1] K. Ramamritham, J.A. Stankovic, and P. F. Shiah, "Efficient Scheduling Algorithms for Real-Time Multiprocessor Systems," IEEE Trans. Parallel and Distributed Systems, vol. 1, no. 2, pp. 184194, Apr. 1990.

[2] Apurva shah,ketan Kotecha,"Adaptive Scheduling Algorithm for real time multiprocessor System",IEEE Advance computing Conference,2009.

[3] Robart Devis,"FPZL Schedulability Analysis", IEEE Real time and embedded Technology and Application Symposium,2011.

[4] Radhakrishna Naik, Vivek Joshi , R.R. Manthalkar," IUF Scheduling Algorithm for improving schedulability,predictability and sustainability of the real time system", Second International Conference on Emerging Trends in Engineering and Technology, ICETET-2009.

[5] Radhakrishna Naik, R.R. Manthalkar, Mukta Dhopeshwarkar" Modified IUF Scheduling Algorithm for Real time Systems", Third International Conference on Emerging Trends in Engineering and Technology,2010 .

[6] Annie s. Wu,han yu,kuo chi lin."An incremental Genetic Algorithm Approach to multiprocessor Scheduling",IEEE Transaction on Parallel and Distributed System,2004.

[7] S. R. Vijayalakshmi,Dr. G. Padmavathi,"A Performance study of GA and LSH in multiprocessor Job Scheduling,International Journal Of Computer Science,2010

[8] Geoffery Black,Ronald Dreslinski,Trevor Mudge,"A Survey of Multicore Processors",2009.

[9] K. Ramamritham, J.A. Stankovic, and P. F. Shiah, "Efficient Scheduling Algorithms for Real-Time Multiprocessor Systems," IEEE Trans. Parallel and Distributed Systems, vol. 1, no. 2, pp. 184194, Apr. 1990.

[10] J. Carpenter, S. Funk, et al. A categorization of real-time multiprocessor scheduling problems and algorithms. In J. Y. Leung, editor, Handbook on Scheduling Algorithms, Methods, and Models, page 30.130.19 , 2004.

[11] K. Lakshmanan, Rajkumar,"Scheduling Parallel Real time Tasks on Multicore Processors",Real Time System Symposium,IEEE 2010.

[12] Fanxin Kong,yang yi,qingxu deng,"Energy Efficient Scheduling of Real time tasks on Cluster based Multicores",2011.

[13] Jinkyu lin,Insik Shin,"LLF Schedulability Analysis on multiprocessor System",Real time system symposium,2010.

[14] Geoffrey Blake, Ronald G. Dreslinski, and Trevor Mudge,"A survey of multicore Processors",November 2009.

[15] Mostafa R. Mohamed, Medhat H. A. Awadalla,"Hybrid Algorithm for Multiprocessor task scheduling", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011.

[16] Oscar H. Ibarra and Chul E. Kim," Heuristic Algorithms for scheduling Independnt tasks on non-identical processors", Journal of association for computing machinery,vol 24,no 2, April 1977,pp 280-289.

[17] Dan McNulty, Lena Olson, Markus Peloquin,"A comparision of scheduling algorithms for multiprocessors", December 2010.

[18] Sanjoy K. Baruah, Member, IEEE, and Joe¨ l Goossens," Rate-Monotonic Scheduling on Uniform Multiprocessors", IEEE Transactions on Computers, VOL. 52, NO. 7, JULY 2003.

[19] Jia Xu," Multiprocessor Scheduling of Processes with Release Times, Deadlines, Precedence, and Exclusion Relations" IEEE Transactions on Software Engineering, VOL. 19, NO. 2, FEBRUARY 1993 .